Amendments to the Claims

1. (Previously Presented) A method of generating a random number, comprising:

sampling data transmitted over a number of microprocessor buses at inputs of a number of multiple input shift registers (MISRs) coupled with the number of microprocessor buses;

generating values within the MISRs based on the sampled data;

retrieving the values from the number of MISRs; and

generating a random number which is based on the values retrieved from the number of MISRs.

2. (Original) The method of claim 1, wherein the number of MISRs is one.

3. (Previously Presented) The method of claim 1, wherein the inputs of one of the number of MISRs are coupled to a data bus which transfers data between a data cache and a CPU core.

4. (Previously Presented) The method of claim 1, wherein the inputs of one of the number of MISRs are coupled to a data address bus which transfers data addresses between a data address cache and a CPU core.

5. (Previously Presented) The method of claim 1, wherein the inputs of one of the number of MISRs are coupled to an instruction data bus which transfers instructions between an instruction data cache and a CPU core.

6. (Previously Presented) The method of claim 1, wherein the inputs of one of the number of MISRs are coupled to an instruction address bus which transfers instruction addresses between an instruction address cache and a CPU core.

7. (Previously Presented) The method of claim 1, wherein the inputs of one of the number of MISRs are coupled to a bus which runs wholly within an integrated circuit package.

8. (Original) The method of claim 1, wherein retrieving values from the number of MISRs comprises:

loading bits of a value stored in a first of the number of MISRs, in parallel, into a temporary register; and

retrieving the value stored in the temporary register.

9. (Original) The method of claim 1, wherein retrieving values from the number of MISRs comprises retrieving a value from a first of the number of MISRs by stepping the first of the number of MISRs to serially shift a plurality of bits out of the MISR.

10. (Original) The method of claim 1, wherein generating a random number comprises hashing together the values retrieved from the number of MISRs.

11. (Original) The method of claim 1, wherein generating a random number comprises XORing the values retrieved from the number of MISRs.

12. (Previously Presented) The method of claim 1, further comprising, prior to randomly sampling data transmitted over the number of microprocessor buses, turning on and initializing each of the number of MISRs upon boot of a computer in which the MISRs reside.

13. (Original) The method of claim 1, wherein value are retrieved from the number of MISRs via an operating system call.

14. (Original) The method of claim 13, wherein said operating system call is of a highest privilege level.

15. (Original) The method of claim 13, wherein generating a random number is performed immediately after the number of MISR readings are taken, the method further comprising storing the random number in a temporary location for subsequent use.

16. (Original) The method of claim 13, wherein said operating system call is issued in response to an application's request for a random number.

17. (Original) The method of claim 1, wherein retrieving values from a number of MISRs comprises a computer program's issuance of a request to read the number of MISRs.

18. (Original) The method of claim 1, wherein generating said random number comprises providing the values retrieved from the number of MISRs, as well as historic values retrieved from the number of MISRs, to a pseudo-random number generator.

19. (Original) The method of claim 1, further comprising testing the number of MISRs by:

initializing the number of MISRs to known values;

executing a test program on the microprocessor in which the number of MISRs reside;

retrieving values from the number of MISRs;

comparing the values retrieved from the number of MISRs with expected values; and

indicating a failure of one of the number of MISRs if its retrieved value does not agree with its expected value.

20. (Previously Presented) The method of claim 1, further comprising, using the random number as an encryption key.

21. (Original) The method of claim 1, wherein the MISRs form part of a microprocessor's built-in self-test hardware.

22. (Original) A method of generating an encryption key, comprising:

assigning a built-in self-test (BIST) local block of a microprocessor a major address;

assigning each of a number of multiple input shift registers (MISRs) in the BIST local block a minor address:

issuing an instruction to turn on and initialize the MISRs;

issuing a request to read the MISRs, in response to a request for an encryption key;

XORing the MISR readings with each other, and with historical readings, if any, to generate an encryption key.